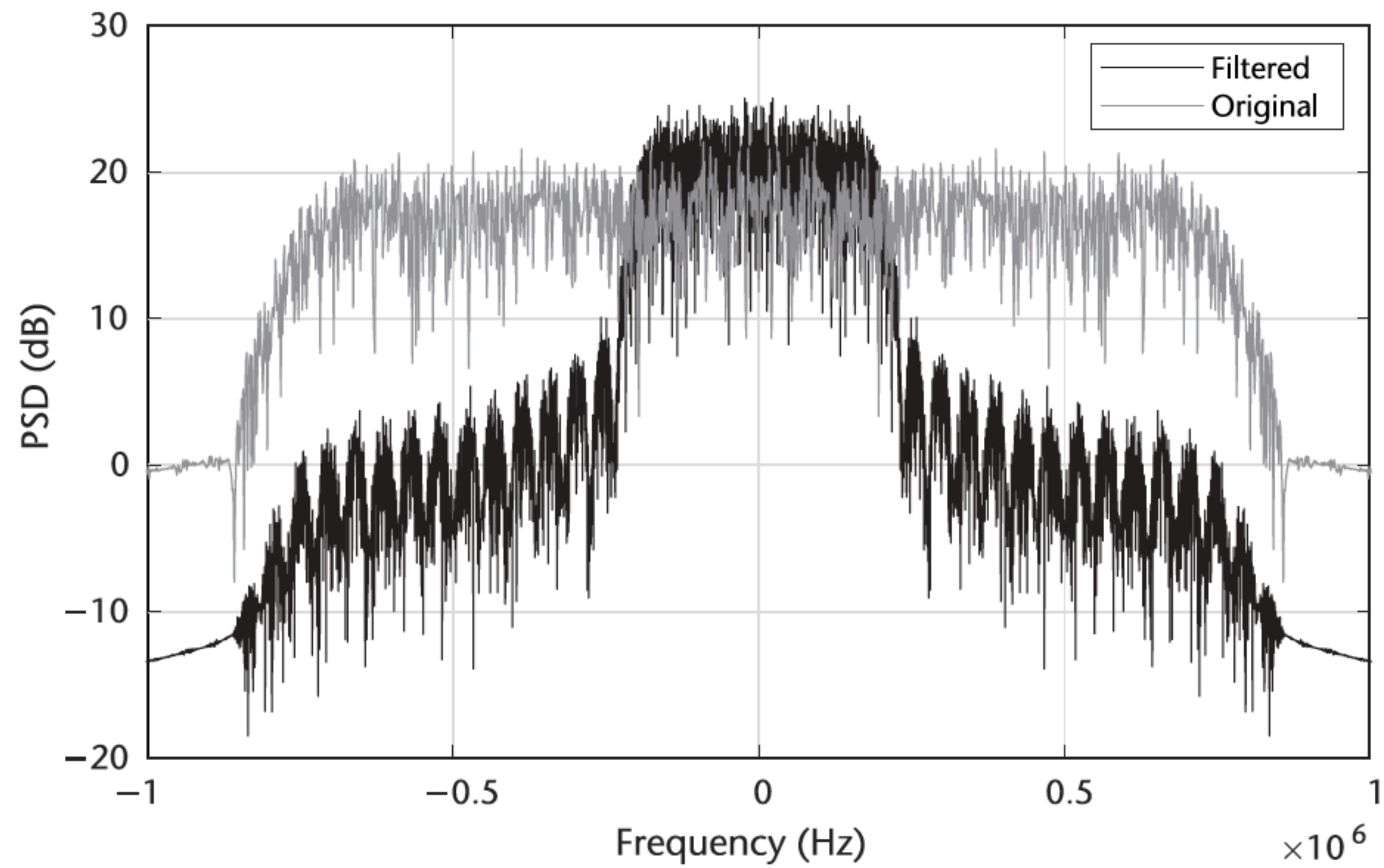
EEU4C21/CSP55031/EEP55C26:
Open Reconfigurable Networks

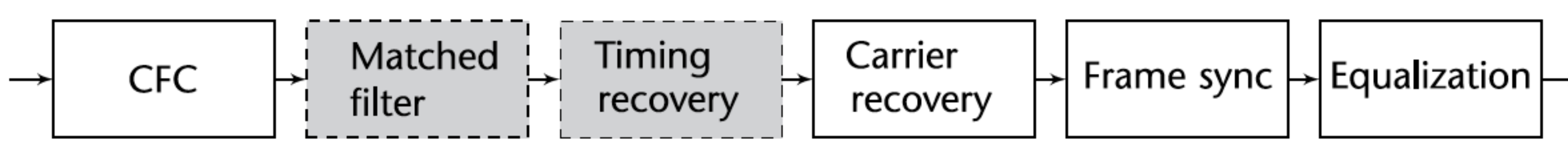
Synchronisation, Estimation, OFDM

Prelude:

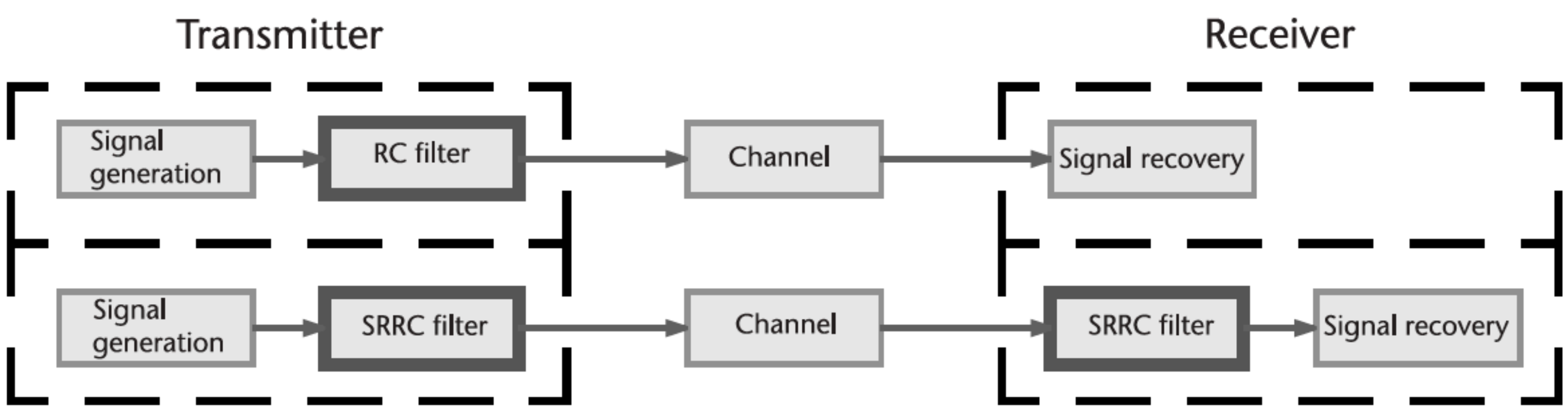
- Recall that we app
band of interest
  - Filtering a symb
  transitions
    - Most filters a
    effective ban
    time-widow (



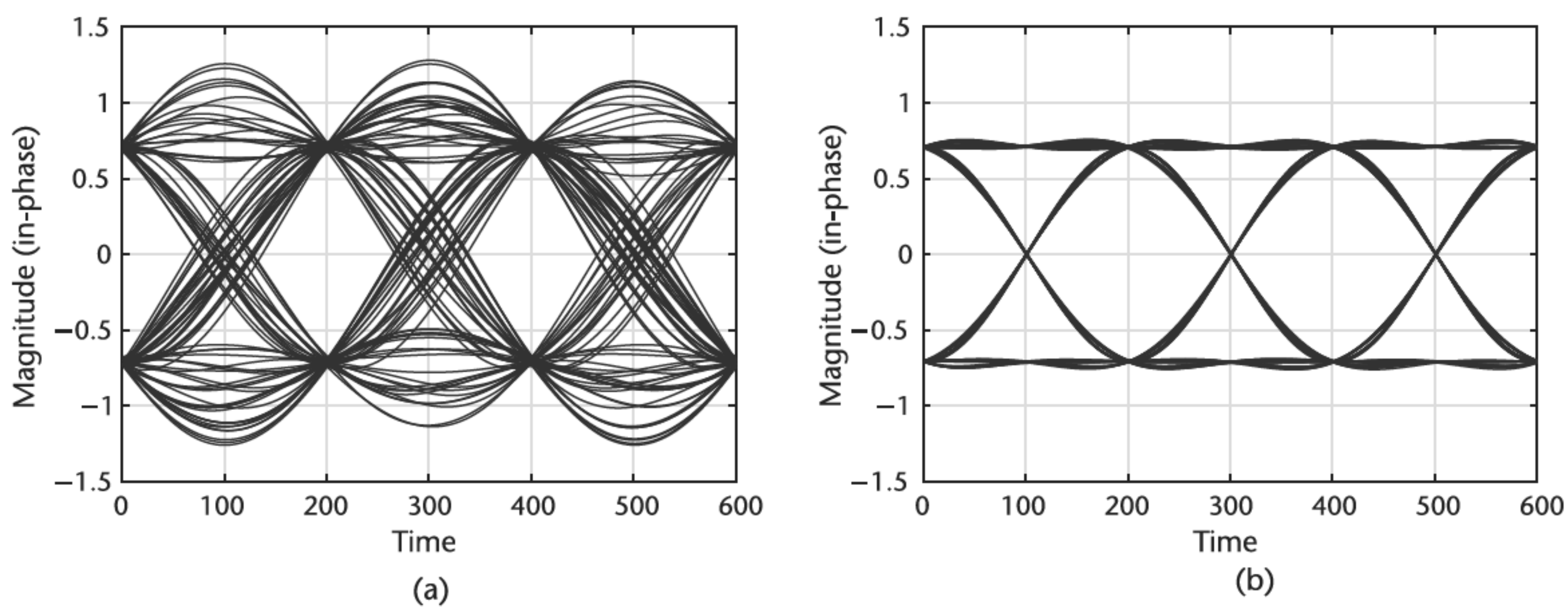
- Tradeoff recoverability for bandwidth - an example filter block used to do this effectively is the square-root raised cosine filter block



Prelude:



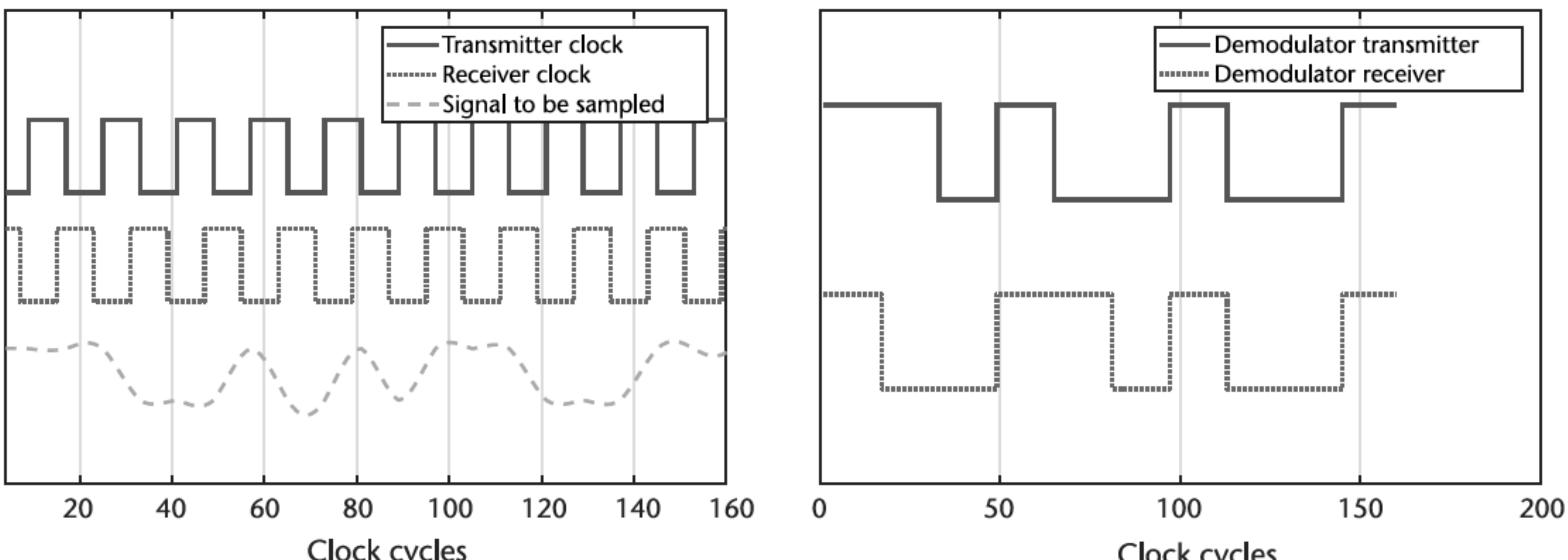
Arrangement of Tx Filters using RC or SRRC filters



Eye diagram of I component of QPSK after applying SRRC at different roll-off factors $\beta = [0.3, 0.99]$

Timing Synchronisation

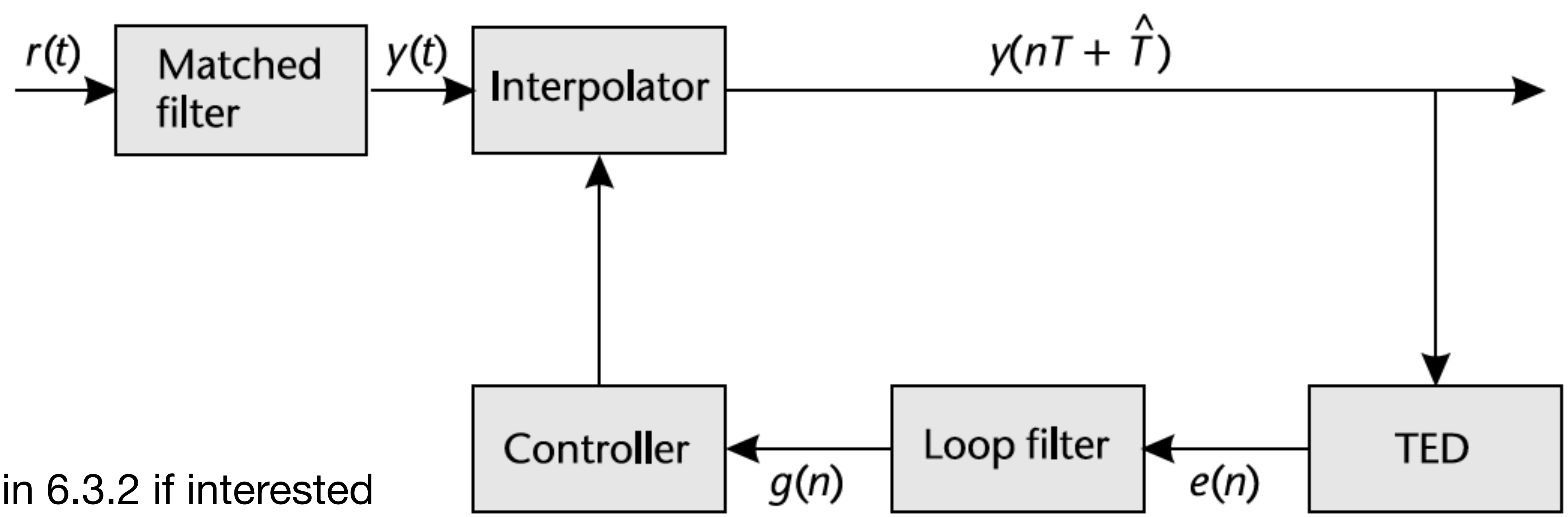
- Purpose of time sync is to align sampling instance between transmitter and receiver - recover the symbol as transmitted
  - Drift in oscillators causes a small delay $\tau <$ clock cycle (fractional



More complex when you have Rx and Tx SRRC/RC filter blocks - see example **`code 6.1`** in text for a simulation you can do with your Pluto devices in loopback mode

Timing correction

- Strategies typically involve timing recovery using phase-locked loops to derive a feedback timing correction
  - Different detection approaches like Zero-Crossing, Muller, Gardner can be applied to detect timing errors (TED block) and correct errors on the fly
  - General working: estimate an unknown offset error, scale the error proportionally, apply an update for future symbols to be correct



See detailed discussion in 6.3.2 if interested

Carrier synchronisation

- Recall that transmitter and receivers are usually spatially separated and distinct nodes
- Their individual local oscillators (LO) will exhibit *relative* frequency offsets due to number of factors - impurities, noise, temperature, age etc and could manifest as random phase noise, frequency offset, drift, and initial phase mismatches (or combinations)
    - We saw this in our lab experiments and applied a course frequency offset correction using the **`ppm`** setting on the Rx Pluto
- Frequency offset for commercial oscillators is specified in PPM - for Pluto, the internal LO is rated at 25 ppm
    - i.e., maximum offset Δf from our operating centre frequency f_c is given by $f_{o,max} = \dfrac{f_c \times PPM}{10^6}$: specifies the recovery design criterion or operational range
    - if $\Delta f > f_{o,max}$ then we should apply other techniques like scanning, not correction

Carrier synchronisation

- Mathematically, we can model corrupted signal at baseband ($s(k)$) with carrier offset ω_o as

$r(k) = s(k)e^{j\omega_o kT+\theta} + n(k)$ where $n(k) \rightarrow$ AWGN noise, $\theta \rightarrow$ carrier phase

- Carrier frequency/phase recovery thus aims to provide a stable constellation for demodulation at the output of synchroniser i.e.., you can estimate the phase difference to detect the frequency offset since $\omega = \dfrac{d\theta}{dt}$

- Typically CFO estimation is applied as a two stage process - coarse correction (CFC estimation) followed by a fine correction (PLL-driven)

Frequency Correction

- CFC can be applied as data-aided (using a correlation type structure) or non-data-aided (or blind) - DAs are restricted to length of preambles although very accurate
    - An example NDA implementation is to use a bin-based correction (though FFT) to detect bin with most energy — corresponding to the tone/frequency (see 7.2.1 for details) -

$$\hat{f}_o = \frac{1}{2TK} \arg \left| \Sigma_{k=0}^{K-1} r^M(k) e^{-j2\pi kT/K} \right|$$

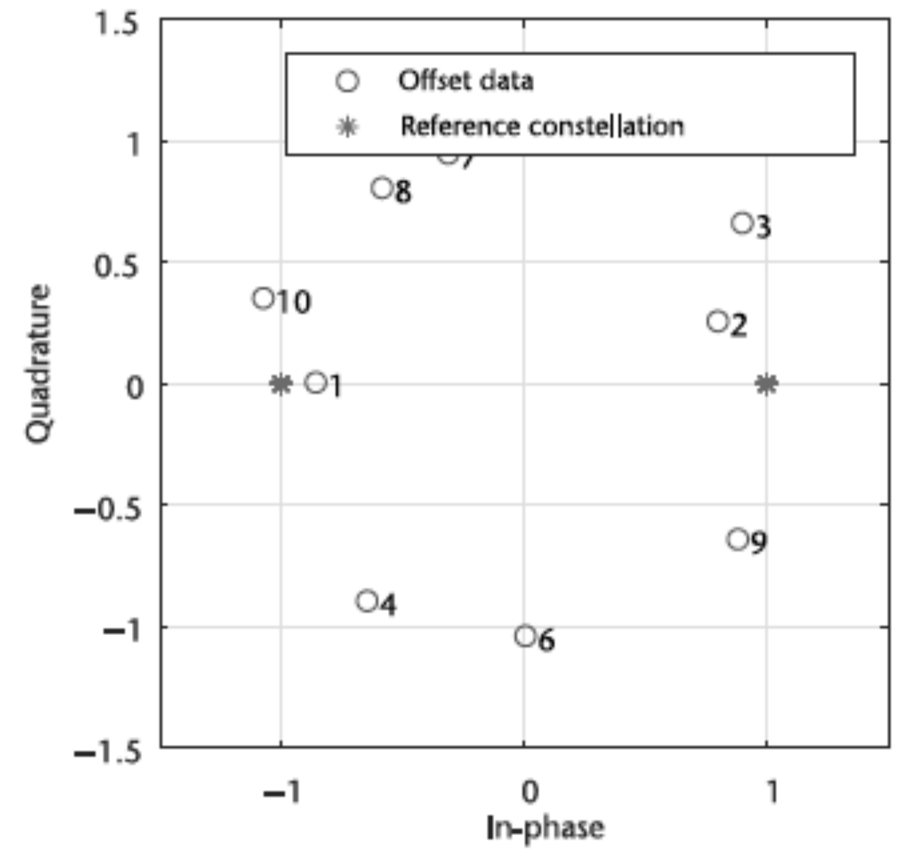
where

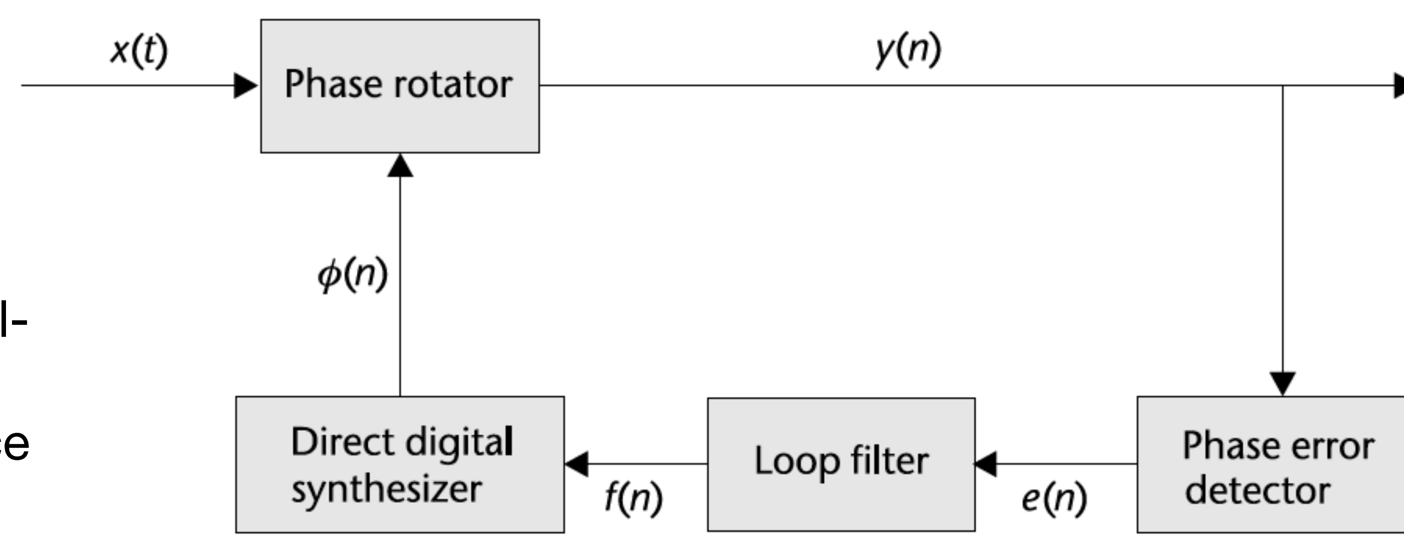
$$K \rightarrow \text{FFT length}, M \rightarrow \text{Modulation order}, f_r = \frac{1}{MTK} \rightarrow \text{Frequency resolution}$$

Note that in our lab, we should have used ppm corrections by estimating this offset and then determining the ppm to apply [a highly stable known pilot tone required]

Fine-frequency correction

- Post coarse correction, offset related to phase might remain, based on the resolution f_r
  - FFC attempts to revert rotational effects caused to the sample constellation (BPSK shown here), driving the received signals frequency offset to zero
  - Note that timing correction is mostly required for this to be visualised correctly from the constellation diagram
  - We use the all-digital PLL based approach, measuring the phase offset using a phase error detector and using its output as the error signal (see 7.2.2 for more details)





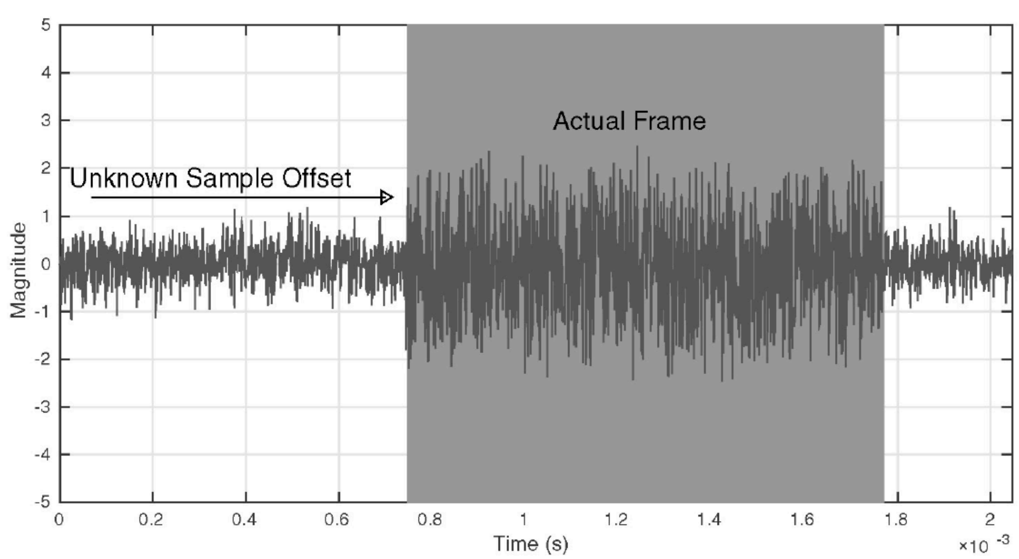
Performance metrics like lock time, effective pull-in range and converged error vector magnitude are used to evaluate synchronisation performance

Phase ambiguity

- While FFC attempts to retrieve the correct orientation of the symbol, there might be ambiguity in the true orientation due to symmetrical modulation schemes and Rx being blind to true orientation of Tx signal i.e., a number of convergent orientations are possible
- Use of code words can aid in resolution of phase ambiguity, by using a known sequence in received data
  - Can be performed pre (computing the offset directly from received preamble symbols) or post demodulation (comparing against orientation mapping expected)
- Differential encoding is another technique, where the true data is depended on the difference between successive bits, not on the bits themselves (i.e., phase invariance is built in)
- Equalizers can also be employed to correct this ambiguity using channel training data (more in Channel estimation)

Frame Synchronisation

- Once timing and frequency offsets are corrected, next we need to identify where the data frame is in the received (demodulated) signal
- Typical method involves appending *preamble* sequences to frames before modulation at Tx
  - The sequences are known exactly at receiver and has specific properties to make accurate estimation possible through cross correlation
  - Common sequences utilised are Barker codes (**`comm.BarkerCode`**) - unique autocorrelation properties with minimal (or in many cases, ideal) off-peak correlation i.e., $|c(k)| = |\Sigma_{i=1}^{N-k} a(i)a(i+k)| \leq 1, 1 \leq k < N$



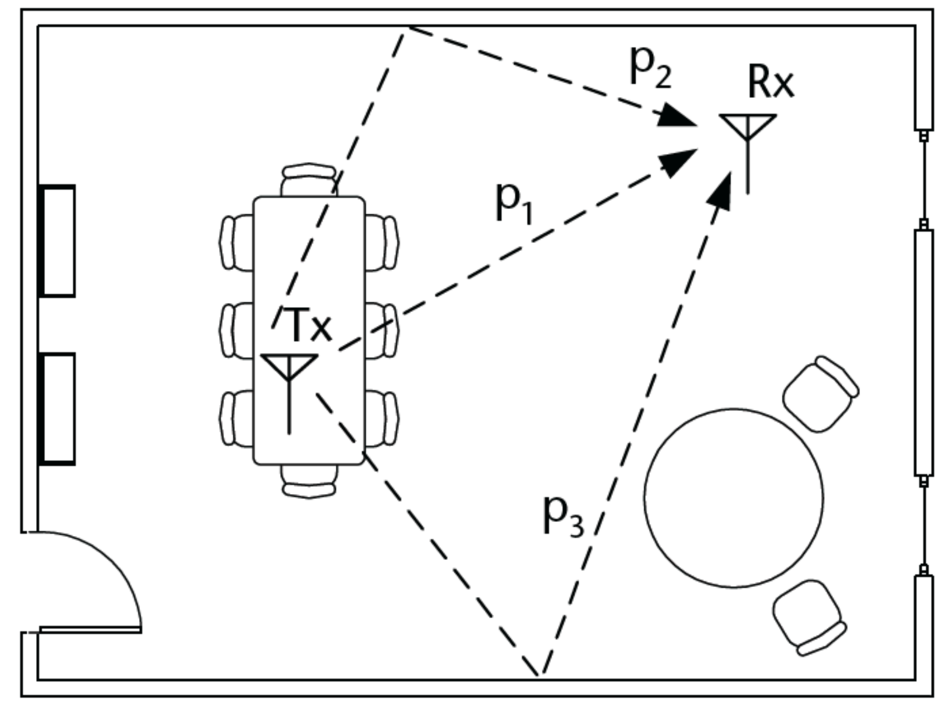
But how would you know if the frame exists in the correlation?

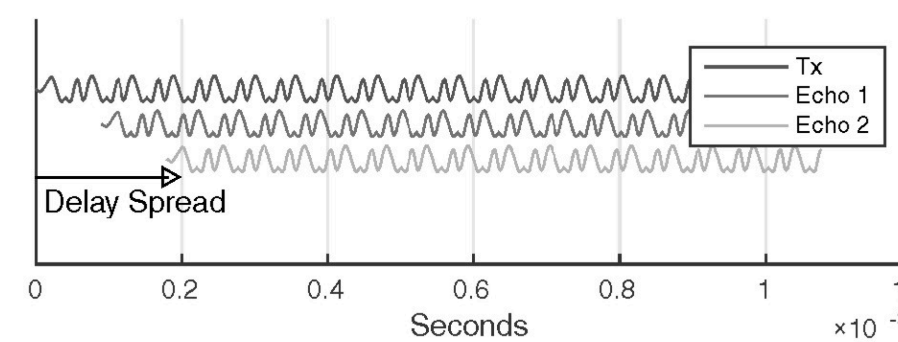
see 8.2.1 (HINT: Thresholding)

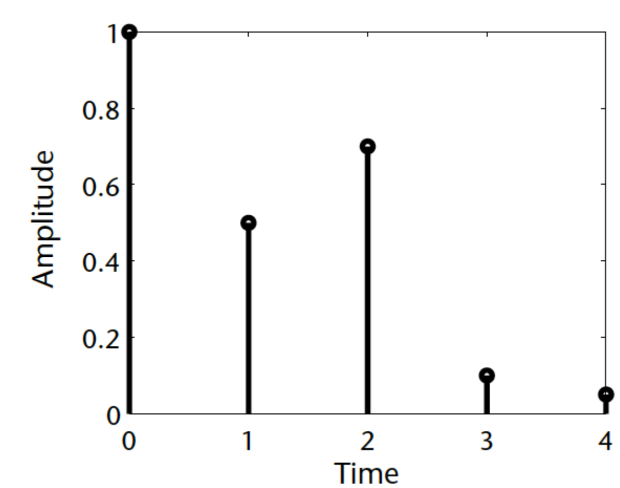
see 8.4 for channel coding schemes

Channel estimation

- What if you receive multiple copies of the same symbol?
  - Multipath caused to to scaled reflections along different paths between Tx and Rx results in echos with different attenuations, delays and frequency/phase offsets
  - Delay spread (D) is related to environment and cause a sample delay $t_s = \frac{B \times D}{c}$; however closely spaced channels can cause large number of scatterers
  - Can be visualised as the channel having an impulse response which varies in aptitude with time
  - Channel estimation 'estimates' the corruption of received data using known information or symbols (training symbols) in the transmitted sequence using algorithms like LMS or RLS



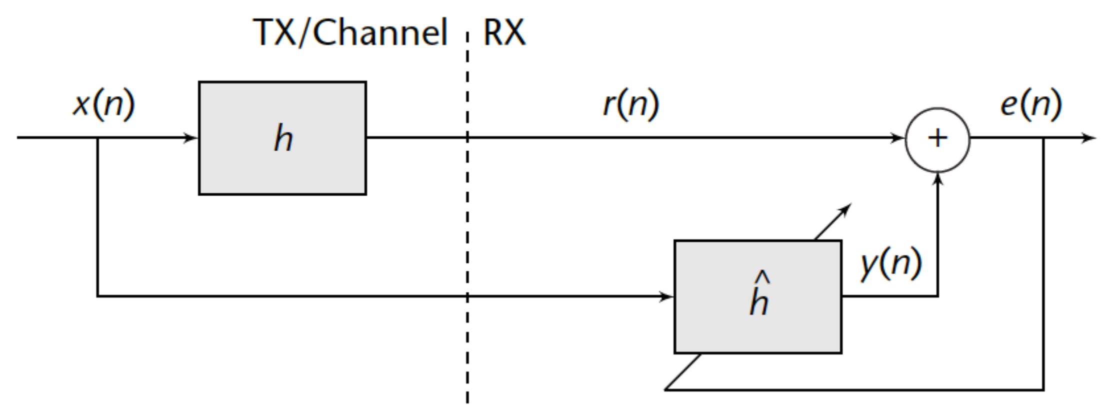




Least Mean Square (LMS) and Recursive Least Squares (RLS); latter is an adaptive filter algorithm

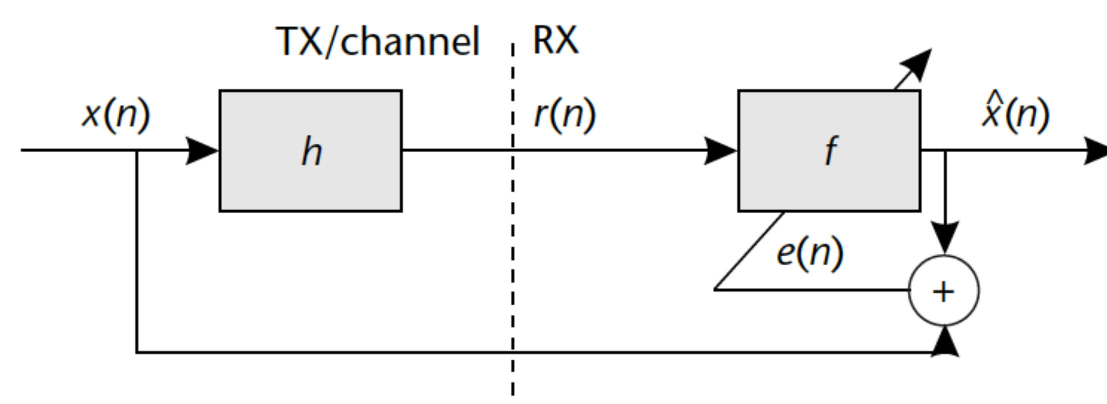
Channel estimation

- Channel estimation: Let h be the channel model (filter response) - the training sequence $x(n)$ tunes our filter $\hat{h}$ to match the channel model



- Equalisers on the other hand uses the knowledge of source data to train an equaliser model that undoes the effects of the channel and remove interference if possible

    - i.e., train f using the training sequence $x(n)$ such that $x(n)$ * h * f $= \hat{x}(n - \delta)$, where δ is the group delay of the cascaded channel filter and equaliser



- More complex realisation of Estimators and Equalisers are possible (chapter 9 for reference)

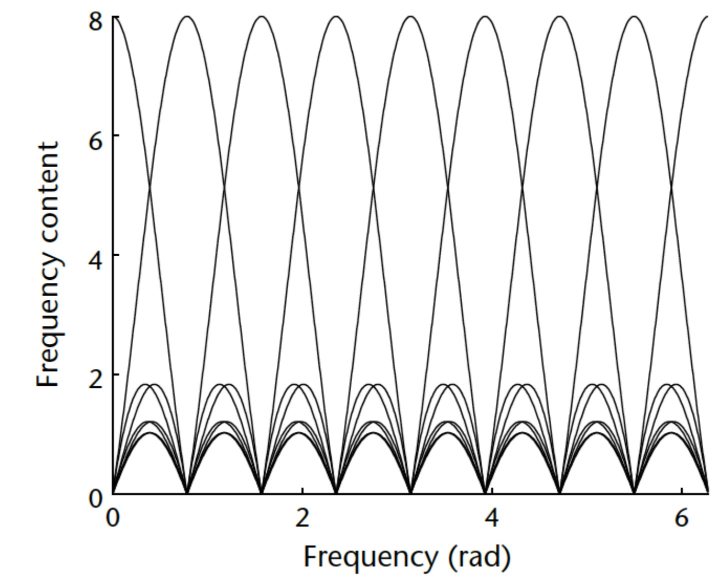
OFDM

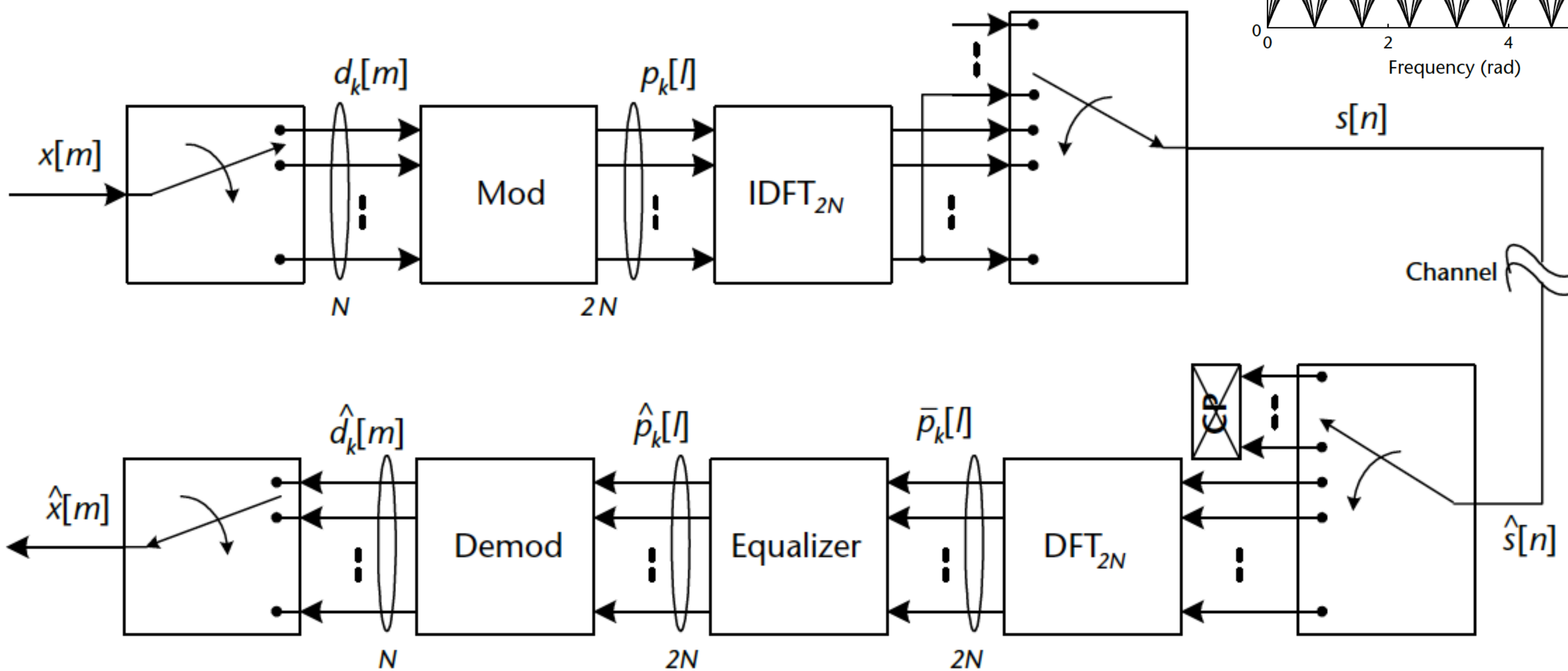
- So far, we used modulation schemes where the sample data modulates a single carrier with frequency f_c
- Multi-carrier Modulation (MCM) multiplexes serial input data into several parallel streams and transmits them over independent sub-carriers (instead of a single carrier)
    - Further, each sub-carrier can be individually modulated and manipulated to optimise for data and/or channel - this ability is particularly useful for combating frequency-selective fading channels
- Hence most high-data-rate communications use some form of MCM, the most popular being Orthogonal Frequency Division Multiplexing

General model

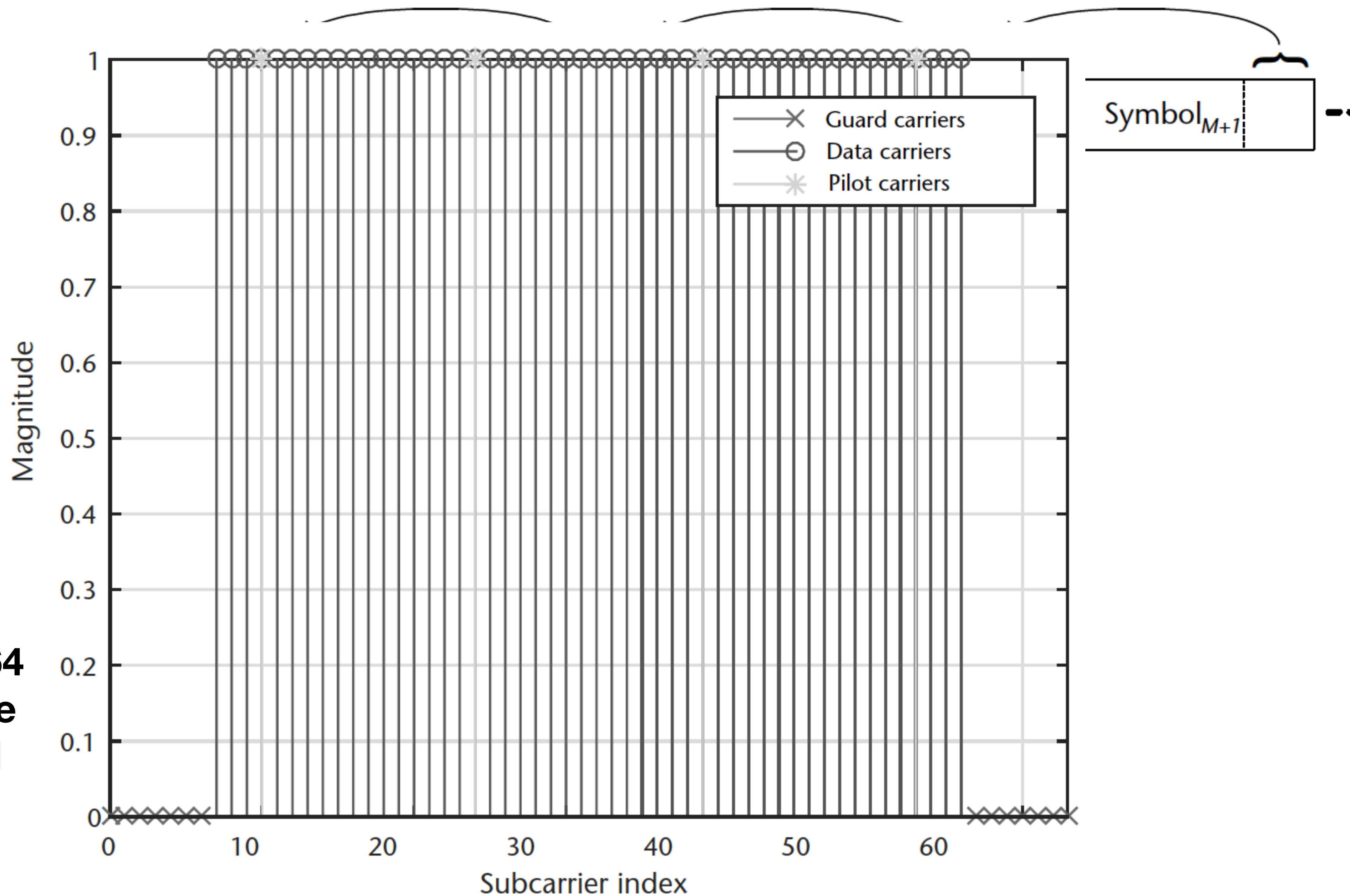
- In essence, OFDM uses inverse DFT (IDFT) and DFT to modulate and demodulate parallel data streams
- At transmitter, digital input $d[m]$ is demultiplexed into N sub-carriers using a demultiplexer (commutator)
    - Each are then modulated into a M-QAM symbol mapping a group of $\log_2(M)$ bits at a time i.e.,$p_k[l] = a_k[l] + jb_k[l]$ for k^{th} subcarrier
    - The 2N-point IDFT combines N different QAM samples to create OFDM symbols - note 2N samples of $s[n]$ constitute an OFDM signal, which is then multiplexed together for transmission
    - i.e., data for transmission is modulated on several sub channels, achieved by multiplying each stream by $\sin(Nx)/\sin(x)$
- The receiver performs the operations in reverse order with 2N-point DFT performing the binning into N sub-carriers

General model





OFDM Wave structure

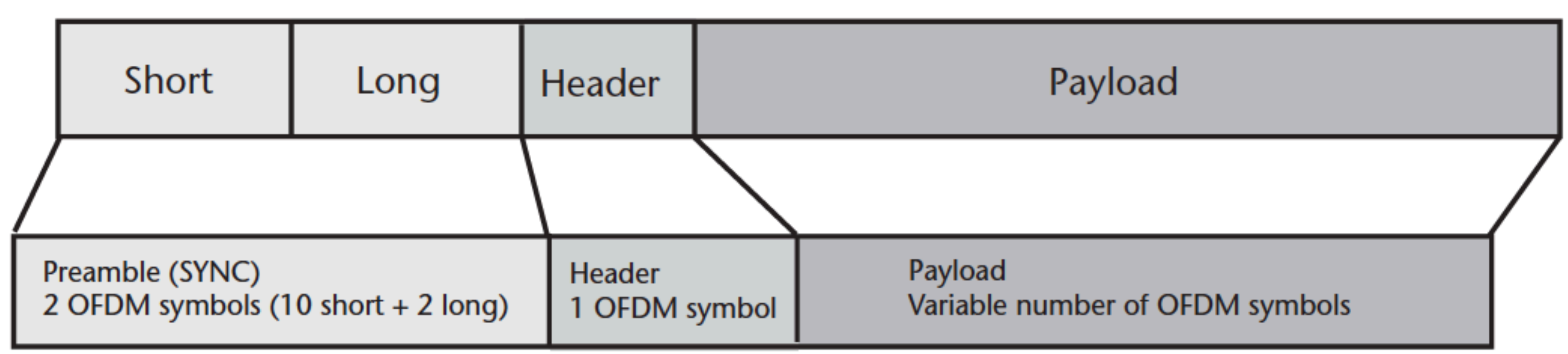


Example of a 64 carrier scheme with 14 guard bands

OFDM Wave structure

- Once OFDM and CP are applied to modulated symbols, additional structure can be added to the frame prior to transmission
  - Such additions are dependent on transmission type - for e.g., continuous transmissions like broadcast or LTE, small insertions are made periodically to enable receiver synchronisations, whereas in burst systems like WLANs, large preamble structures are used
  - The WLAN frame for instance has a preamble with 4 OFDM symbols including short portion LSTF (10 repeated short sequence) and long field LLTF including two repeated sequences and a CP
  - LTE on the other hand uses periodic Primary and Secondary Synchronisation Signals (PSS, SSS) along with pilot tones for correct reception

Legacy Short Training Field - useful for AGC and CFO estimation
Legacy Long Training Field - preamble for channel estimation



See 10.4 onwards to see details on how Synchonisation, Packet Detection and equalisation are applied in OFDM